

Opis

Dana jest niepusta tablica dwuwymiarowa $a[0][0], \dots, a[n-1][m-1]$ liczb całkowitych, w której każdy wiersz i każdą kolumna jest posortowaną niemalejąco. Napisz w Javie program wyszukujący podany element 'x' w tablicy za pomocą czterech różnych funkcji:

- `recFirst(...)` - funkcji rekurencyjnej, wyznaczającej pierwsze wystąpienie elementu 'x',
- `recLast(...)` – funkcji rekurencyjnej, wyznaczającej ostatnie wystąpienie elementu 'x'
- `iterFirst(...)` - funkcji iteracyjnej, wyznaczającej pierwsze wystąpienie elementu 'x'
- `iterLast(...)` - funkcji iteracyjnej, wyznaczającej ostatnie wystąpienie elementu 'x'.

Pierwsze wystąpienie elementu oznacza najwcześniejsze wystąpienie leksykograficznie (czyli najwcześniejszy wiersz, następnie jak najwcześniejsza kolumna), natomiast ostatnie – najpóźniejsze leksykograficznie.

W ciele funkcji rekurencyjnych nie może pojawić się żaden rodzaj pętli, natomiast funkcje iteracyjne nie mogą wywoływać innej utworzonej funkcji.

Wszystkie funkcje mają mieć średnią złożoność czasową **silnie mniejszą** od złożoności kwadratowej, czyli: $T(n,m) \ll O(\max(n, m)^2)$.

Wejście

Dane do programu wczytywane są ze standardowego wejścia (klawiatury) zgodnie z poniższą specyfikacją. Pierwszą podawaną wartością będzie dodatnia liczba całkowita oznaczająca ilość zestawów danych, po której na wejściu pojawią się zestawy danych w ilości równej wczytanej liczbie.

Każdy zestaw danych zawiera dwie dodatnie liczby całkowite z zakresu od 1 do 100, oznaczające odpowiednio liczbę wierszy oraz liczbę kolumn tablicy, w następnych liniach podawane są dane będące kolejnymi wierszami tablicy zgodnie z podaną liczbą wierszy i kolumn. Dane każdego zestawu są liczbami całkowitymi z zakresu od -2^{15} do $+2^{15}$.

Na końcu zestawu podawana jest wartość 'x', która ma zostać wyszukana w tabeli kolejno za pomocą wszystkich czterech funkcji.

Wyjście

Dla każdego zestawu danych program wypisuje cztery linie, będące efektem działania kolejnych czterech funkcji oraz znak „---” w piątej:

```
recFirst: x = a[i1][j1]
recLast: x = a[i2][j2]
iterFirst: x = a[i3][j3]
iterLast: x = a[i4][j4]
---
```

Litery przed dwukropkiem oznaczają identyfikator wywołanej funkcji, 'x' oznacza szukany element, natomiast kolejne indeksy 'i' oraz 'j' oznaczają wiersz i kolumnę ze znalezionym wystąpieniem elementu. W przypadku braku elementu 'x' w tablicy, po dwukropku ma pojawić się napis "x missing in array".

Wymagania implementacyjne

1. W pierwszej linii program powinien zawierać komentarz: `// Nazwisko i imię – nr grupy`
2. Jedynym dozwolonym importem jest obsługa wczytywania z klawiatury, to jest: `import java.util.Scanner;`
3. Główna klasa musi nazywać się `Source`, co oznacza ogólne ramy kodu postaci:

```
class Source {
    public static void main( String [] args ) {
        ...
    }
}
```

4. Wczytywanie musi się odbywać przez pojedynczą zmienną skanera wczytywania, zadeklarowaną zewnętrznie w stosunku do wszystkich metod głównej klasy.
Oznacza to tylko jedną deklarację w przykładowej postaci:
`public static Scanner in = new Scanner(System.in);` w pierwszej linii ciała głównej klasy.

5. Program powinien zawierać komentarze:

- (a) opisujące ideę działania wszystkich czterech funkcji w programie wraz z uzasadnieniem ich złożoności
- (b) objaśniające działanie istotnych fragmentów kodu w funkcji
- (c) opisujące znaczenie ważniejszych zmiennych w funkcjach

6. Na końcu pliku źródłowego proszę podać w komentarzu przykłady własnych danych wejściowych i wyjściowych, sprawdzających wszystkie przypadki występujące w programie, dla których był uruchomiony program przed wysłaniem na BaCę.

Przykład danych

Wejście:	Wyjście:
2	recFirst: 20 = a[1][1]
3 4	recLast: 20 = a[2][2]
10 10 10 10	iterFirst: 20 = a[1][1]
10 20 20 30	iterLast: 20 = a[2][2]
20 20 20 40	---
20	
3 4	recFirst: 50 missing in array
10 10 10 10	recLast: 50 missing in array
10 20 20 30	iterFirst: 50 missing in array
20 20 20 40	iterLast: 50 missing in array
50	---